Derin Öğrenme Algoritmaları ile Covid-19 Analizi

Abdulaziz ÖZ
Bilgisayar Mühendisliği Bölümü, Düzce Üniversitesi, Düzce, Türkiye
abdulaziz.oz@outlook.com
Furkan ÇINAR
Bilgisayar Mühendisliği Bölümü, Düzce Üniversitesi, Düzce, Türkiye
furkancinar46@gmail.com

ÖZETÇE

Yapılmış olan bu çalışmanın amacı son zamanlarda çıkan koronavirüs hastalığının uzun kısa süreli bellek, eğri uydurma ve çift yönlü uzun kısa süreli bellek algoritmalarıyla analizi yapıldıktan sonra algoritmaların karşılaştırılmasıdır. Karşılaştırma yaparken Türkiye'deki koronavirüs hastalığının ölüm, iyileşen sayısı ve hasta sayısı verileriyle oluşturulan bir veri seti üzerinden inceleme yapılmıştır. Araştırmada ilk başta kullanılmış olan algoritmalar, daha sonra ise kullanılmış olan optimizasyon metrikleri anlatılmıştır. Deney sonucu algoritmaların birbirleriyle sonuçları karşılıklı olarak incelenmiştir. Hangi algoritmanın daha iyi sonuç verdiğine ulaşılmaya çalışılmıştır.

ABSTRACT

The aim of this study is to compare the algorithms after analyzing the recent coronavirus disease with long short term memory, curve fitting and bidirectional long short term memory algorithms. While making comparisons of coronavirus cases in Turkey; The study was conducted on a data set consisting of death, the number of recovered and the number of patients. In the study, the algorithms used at first and then the optimization metrics used were explained. As a result of the experiment, the results of the algorithms with each other were examined mutually. It has been tried to find out which algorithm gives better results.

I. GİRİŞ

Koronavirüs (CoV), toplumda yaygın olan soğuk algınlığı, hafif enfeksiyonlardan, Orta Doğu Solunum Sendromu (Middle East Respiratory Syndrome, MERS) ve Ağır Akut Solunum Sendromu (Severe Acute Respiratory Syndrome, SARS) gibi daha tehlikeli hastalıklara neden olabilen büyük bir virüs ailesidir.[1]

Dünya sağlık örgütü Aralık ayında Çin'in Vuhan şehrinde ateş, öksürük nefes darlığı gibi rahatsızlıklardan dolayı olan vakaları bildirmiştir. 1 Aralık 2019 tarihinde ise daha önce insanlarda tespit edilmemiş yeni bir koronavirüs vakası olarak kabul edilmiştir.

Avrupa'da ilk vaka 22 Ocak tarihinde Almanya'da görülmüştür. Dünya Sağlık Örgütü, Çin dışında 113 ülkede Covid-19 vakalarının görülmesinden dolayı 11 Mart tarihinde Covid-19 salgınını pandemi olarak tanımlamıştır. Ülkemizde ise ilk vaka 11 Mart tarihinde görülmüştür. Yapılmış olan bu araştırmada ise 22 Ocak 2020 tarihinden 19 Ağustos 2020 tarihine kadar olan Covid-19 Türkiye Hasta, ölüm ve iyileşen sayıları incelenmiştir.

19 Ağustos tarihine kadar olan Covid-19 hasta sayısı 253108, ölüm sayısı 6039, iyileşen sayısı ise 233915 olarak tespit edilmiştir.

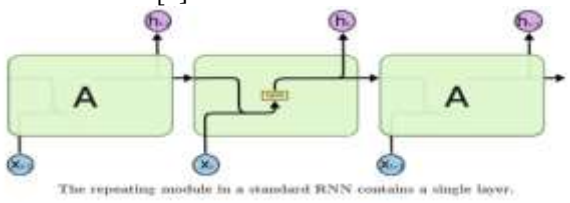
Elde edilen bu verilerin incelenmesi için Uzun Kısa Süreli Bellek (Long Short-Term Memory, LSTM), Çift yönlü Uzun Kısa Süreli Bellek (Bidirectional Long Short-Term Memory, Bi-LSTM) ve Eğri Uydurma (Curve Fitting) yöntemleri kullanılmıştır.

II. DERİN ÖĞRENME MİMARİLERİ

Bu çalışmada derin öğrenme algoritması olan LSTM, Bi-LSTM ve Curve Fitting yöntemleri kullanılmıştır. Bu bölümde bu algoritmaların detayları hakkında bilgiler verilmiştir.

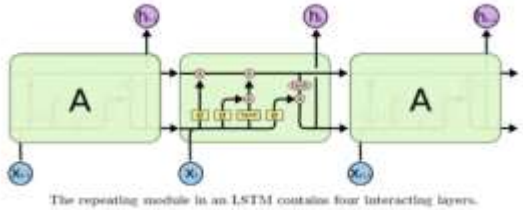
A. Uzun Kısa Süreli Bellek

Uzun Kısa Süreli Bellek(LSTM) uzun süreli bağımlılıkları öğrenebilen RNN'nin bir türüdür. Hochreiter ve Schmidhuber tarafından 1997 yılında tanıtıldı. Çok çeşitli problemler üzerinde çok iyi bir şekilde çalışmaktadırlar ve şu anda yaygın olarak da kullanılmaktadır.[4]



Şekil 1. Tek katmanlı standart RNN

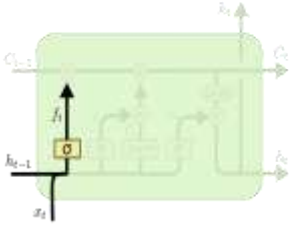
LSTM'ler ardı ardına birbirini takip eden yapıya sahiptir. Lakin bir sonraki parçanın yapısı daha farklıdır. Etkileşimli olan dört parçalı sinir ağı katmanına sahiptir.



Şekil 2. Dört Etkileşim katmanı olan LSTM

- Birinci adım

İlk adımda, ağ gereksiz olan bilginin tespitini yapar ve hücreden atma işlemini belirler. Şekil 5'te hücreden atılma işlemi kararını vermek için unutma geçidi katmanı olarak bilinen sigmoid fonksiyonu katmanı gösterilmektedir.[3,4]

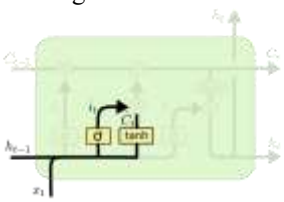


$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \; + \; b_f\right)$$

Şekil 3. Unutma geçidi katmanı

- İkinci adım

İkinci adımda, ağda hücre durumuna depolanması gereken bilginin kararı veriliyor. Bütün bu süreç takip adımlarından taviz vermektedir. Girdilerin hangisinin güncellenmesi gerektiğini giriş kapısının katmanı isimli sigmoid katmanı belirler. Bundan sonra, duruma eklenecek olan yeni adayların vektörünü hazırlayan tanh katmanı şekil 6'daki gibi kullanılmaktadır. [3,4]

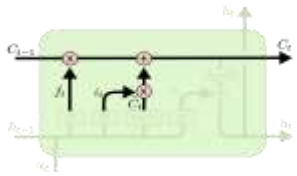


$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C)$$

Şekil 4. Giriş geçidi ve yeni aday değer vektörü

- Üçüncü adım

Üçüncü adımda, C_t değeri eski hücrenin durumu olan C_{t-1} değerinden güncellenir. İlk başta eski durum için unutulması için kararlaştırılan bilgileri unuttuktan sonra f_t ile çarpılır. Bundan sonra, çıkan sonuçların güncellenmesi kararlaştırıldıktan sonra yeni adayların değeri olan $i_t \, x \, C_t$ eklenir. [3,4]

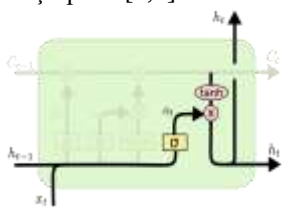


$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

Şekil 5. Yeni hücre durum

- Dördüncü adım

Sonuncu adım için, hangi bölümün çıktı olduğuna karar verdikten sonra sigmoid katmanı çalıştırılır. Daha sonra, ihtiyacı olan parçalara çıktı olarak ulaşmak için hücre durumu tanh'ten geçirilip sigmoid kapısındaki çıkan değer ile çarpılır. [3,4]



$$o_t = \sigma\left(W_o \left[h_{t-1}, x_t\right] \; + \; b_o\right)$$
$$h_t = o_t * \tanh\left(C_t\right)$$

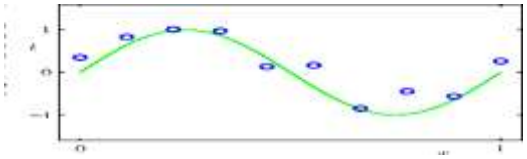
Şekil 6. Çıkış geçidi ve yeni bilgi

B. Eğri Uydurma

Genellikle Veriler bir süreklilik müddetince farklı değerler için verilmektedir. Fakat, farklı değerlerin olduğu noktalar için tahmin yapmak gerekirse eğri uydurma yöntemleri bu işlem için uygulanmaktadır. Herhangi bir fonksiyon karmaşık halinden basitleştirilmiş haline ihtiyacımız olduğunda da eğri uydurma yöntemi kullanılabilir. Bunu ilgilenilen aralığın içindeki farklı noktalarda olan fonksiyonun değerlerini hesaplayarak ulaşılabilmektedir. Böylece bu değerlerle uyuşan basitleştirilmiş bir fonksiyon üretilebilir.

Eğri uydurma iki yaklaşım ile incelenmektedir. İlk yaklaşım, verilerin yüksek derecede hatalı olduğu yerde yöntem, verilerin eğilimini tek bir eğri ile göstermektir. Eğri tasarlanırken gruplanan noktaların genel şekline göre oluşturulur. En küçük kareler yöntemi bu yaklaşımlardan biridir.

İkinci yaklaşımda ise, çok hassas verilerin olduğu yerlerde noktaların üzerinden bir eğri oluşturmaktır. Böyle veriler çoğunlukla tablolardan oluşmaktadır. Ayrık noktaların tahmini için kullanılan bu yöneme interpolasyon adı verilmiştir.

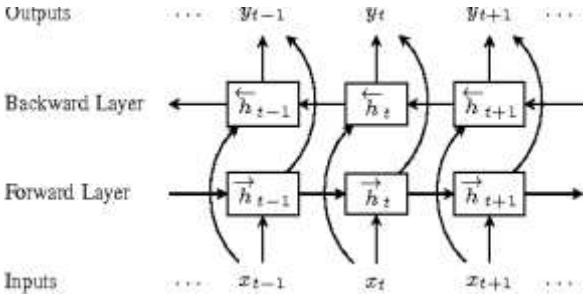


Şekil 7. Eğri uydurma

C. Çift Yönlü Uzun Kısa Süreli Bellek

Çift yönlü LSTM'ler, sıra sınıflandırma problemlerinde model performansını iyileştirebilen geleneksel LSTM'lerin bir uzantısıdır.

Giriş dizisinin tüm zaman adımlarının mevcut olduğu problemlerde, Çift Yönlü LSTM'ler giriş dizisinde bir LSTM yerine iki tane çalıştırır. Birincisi olduğu gibi giriş dizisi üzerinde ve ikincisi giriş dizisinin ters kopyası üzerindedir. Bu, ağa ek bağlam sağlayabilir ve sorunla ilgili daha hızlı ve hatta daha eksiksiz öğrenmeye neden olabilir.



Şekil 8. Bi-LSTM Modeli

Ağdaki ilk tekrarlayan katmanın kopyalanmasını, böylece artık yan yana iki katman olmasını, ardından giriş dizisinin ilk katmana girdi olarak olduğu gibi sağlanmasını ve giriş dizisinin ikinci katmana ters bir kopyasının sağlanmasını içerir. Sekansın iki yönlü olarak sağlanmasının kullanımı, başlangıçta konuşma tanıma alanında gerekçelendirilmiştir.

III. DENEYSEL ANALİZ

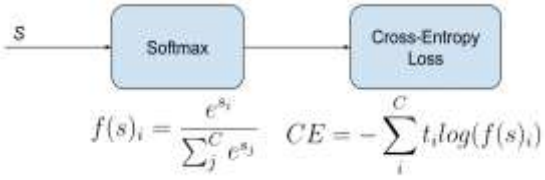
A. Kullanılan Veri Seti

Kullanılan veri setinde 22.10.2020 tarihinden 19.08.2020 tarihine kadar olan günlük Türkiye'de tespit edilmiş olan Covid-19 hastalarının sayısı, Covid-19 hastalığından vefat edenlerin sayısı ve iyileşenlerin sayısı bulunmaktadır.

B. Performans Metrikleri

Oluşturulan modellerde performans metriği olarak doğruluk ve kayıp değerleri hesaplanmaya çalışılmaktadır.

İlk olarak kaybolan değeri bulabilmek için modellerin derlenmesi gerekir. Kayıp değer, tahmin edilen ile beklenen gerçek çıkış arasındaki farkın nicel olarak hesaplanmasıdır. Bu değer, çıkışın tahmin edilen model tarafından yapılmış olan hata ölçümünü vermektedir. Yani kayıp değeri, modelin test edilirken ne kadar düzgün çalıştığını hesaplar. Kaybın düşük olarak hesaplandığı değerler, modelin iyi çalıştığını gösterir. Bu çalışmada, kayıp fonksiyonu olarak kategorik çapraz entropi kullanılmıştır. Softmax Kaybı olarak da bilinir. Bir Softmax aktivasyonu artı bir Çapraz Entropi kaybıdır. Çok sınıflı sınıflandırma için kullanılır.



Şekil 9. CrossEntropy

Uygulamada Sgd, Adam ve Rmsprop ile optimizasyon yapılmıştır. Veri setinin eğitilme süresini belirleyip "fit" yöntemiyle epok sayısına göre eğitilmektedir. model.evualete metoduyla, sonuçlar incelenir. Doğruluk hesaplaması için aşağıdaki formül uygulanmaktadır.

$$Doğrulama = \frac{\text{Doğru öngörülen sınıfı}}{\text{Toplam test sınıfı}} x\ 100 \qquad (1)$$

1) Olasılıksal Dereceli Azalma

Olasılıksal Dereceli Azalma(SGD), metin sınıflandırırken ve doğal dil işleme işlemi yapılırken fazlaca karşılaşılan seyreltilen ve büyük ölçekli olan makine öğrenmesi problemleri için başarılı bir şekilde uygulanmaktadır. Verilerin seyrek olduğu göz önüne alındığında, bu modüldeki sınıflandırıcılar, 10^5'den fazla eğitim örneği ve $10^{5'}den$ fazla özellik içeren problemlere kolayca ölçeklenebilir. Verimlilik ve uygulama kolaylığı açısından avantajlıdır.[2]

$$\boldsymbol{\theta = \theta - \alpha \nabla_\theta J(\theta; x^{(i)}, y^{(i)})}$$

Şekil 10. Güncelleme ifadesi

2) Ortalama Karekök Yayılımı

Ortalama Karekök Yayılımı(RMSProp), Adagrad yönteminin azalmakta olan öğrenim oranını, eğimlerin karesinin hareketli ortalamalarından çözüme ulaşmaya çalışır. Eğimi normalleştirmek için geçmiş eğimlerin inişlerinin büyüklüğünden faydalanmaktadır. RMSProp'ta, her bir parametrenin farklı hızlarda öğrenmektedirler ve bu hızları otomatik seçilir. RMSProp öğrenme oranı, üstel eğimlerin karesinin sönüm ortalamasına bölmektedir.[3]

$$\theta_{t+1} = \theta_t - \frac{n}{\sqrt{(1-y)g_{t-1}^2 + y g_t + \varepsilon}}\ x\ g_t \qquad (2)$$

3) Uyarlamalı Momentum

Uyarlamalı Momentum(Adam), herbir Parametre de uyarlamalı öğrenme oranını, gradyanların 1. ve 2. Momentlerinden tahmini sonuç hesaplayan bir optimizasyon algoritmasıdır. Adagrad'ın azalmakta olan öğrenme oranlarını daha da azaltmaktadır. Adam, RMSprop ile Adagrad birleşimi gibi düşünülebilir. Adam hesaplamalarda daha verimli ve belleğin az kullanımından dolayı en popüler gradyan iniş optimizasyon algoritmalarından biridir.[5]

$$\theta_{t+1} = \theta_t - \frac{n}{\sqrt{\widehat{v_t}+\varepsilon}} \times \widehat{m_t} \qquad (3)$$

C. Kullanılan Araçlar ve Kütüphaneler

Veri setinin modellerken, test ve eğitim yaparken Python diliyle kodlanmıştır. Modelleri oluştururken TensorFlow ve Keras kütüphaneleri kullanılmıştır. Modelleri eğitirken ilk başta veri setinin test ve doğrulama olarak ayırma işlemi gerçekleştirilmiştir. MNIST veri setinde 60000 eğitim, 10000 test görüntüsü bulunmaktadır.

1) Tensorflow

Açık kaynak kodlu bir deep learning kütüphanesidir. Tensorflow kütüphanesi tek bir API ile birden çok platformda kullanılabilmektedir. Mobil uygulama, web uygulaması veya IoT cihazlarda projeler geliştirirken projelerde bu kütüphaneyi kullanım sağlayabiliriz. Hesaplamaları birden çok CPU ve GPU kullanarak uygulamamıza imkân sağlar. Google'ın makine ve derin sinir ağları için geliştirdiği bir kütüphanedir. [6]

2) Keras

Keras Python diliyle yazılan üst düzey bir sinir ağı kütüphanesidir (API). Theano ve Tensorflow üzerine kurulmuştur. Deneyleri daha hızlı sonuca ulaştırabilmek için tasarlanmıştır. Açık kaynaklı, kullanıcılar için çok kolay bir yapıda ve genişletilebilir. Hem CPU hem de GPU üzerinde çalışıp birçok algoritmayı desteklemektedir. Derin öğrenme yöntemlerinde modellerin oluşumunda Tensorflow'a göre daha kolay ve basit bir yapıdadır.[7]

3) Matplotlib

Python matplotlib; matplotlib.pyplot, 2D, 3D grafikler için kullanılır. Bu kütüphane ile verilerin görselleştirmesi sağlanır.

D. Modelleri Oluşturma

Bu kısımda modeller incelenmektedir. Modellerin giriş ve çıkış katmanları, gizli katmanlar, evrişim sayısı, kaç epokta işlemlerin yapıldığı ve bu işlemler sonucu ortaya çıkan kayıp ve doğruluk oranlarının kaç çıktığı bulunmaya çalışılmaktadır.

1) Uzun Kısa Süreli Bellek

LSTM için Loss değeri olarak mse kullanılmıştır. Optimizasyon olarak ise adam kullanılmıştır. 2-katmanlı bir yapı oluşturulmuştur.

Katman	Nöron Sayısı	Seyreltme Katmanı
1.Katman	64	0.2
2.Katman	32	0.2

Tablo 1. LSTM için oluşturulan modelTablosu

2) Eğri Uydurma

Bulaşıcı bir hastalığın yayılmasında lojistik eğri kullanılabilir. Üssel bir şekilde büyümeye başlarken bükülme noktasında yavaşlamaya başlar.Onaylanan vakalar lojistik eğri ile modellenmiştir.

$$f(x) = \frac{N}{1+e^{-k(x-x_0)}}$$

Şekil 11. Lojistik Eğri Formulü

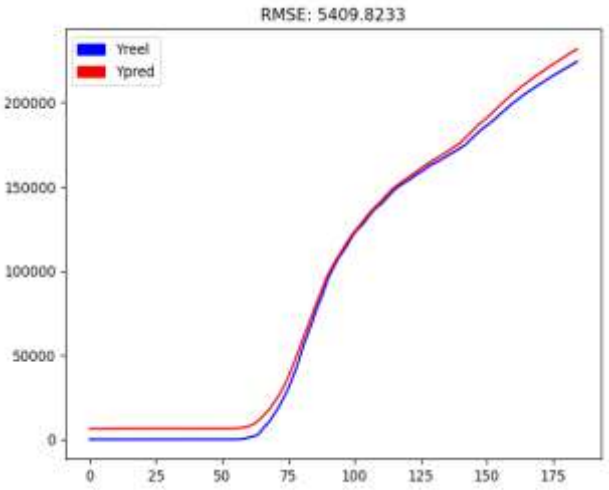
3) Çift Yönlü Uzun Kısa Süreli Bellek

Bi-LSTM için Loss değeri olarak mse kullanılmıştır. Optimizasyon olarak ise rmsprop kullanılmıştır. 2-katmanlı bir yapı oluşturulmuştur.

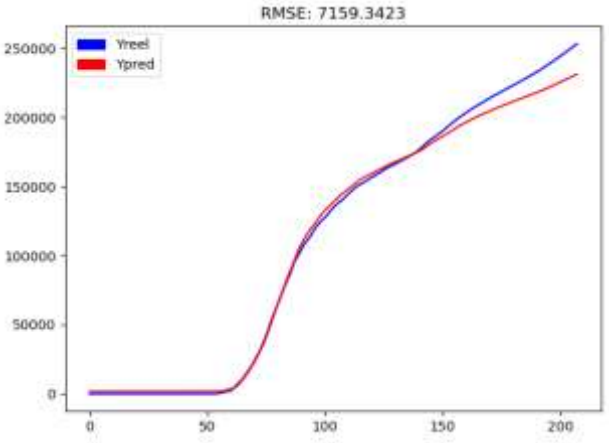
Katman	Nöron Sayısı	Seyreltme Katmanı
1.Katman	64	0.2
2.Katman	32	0.2

Tablo 2. Bi-LSTM için oluşturulan modelTablosu

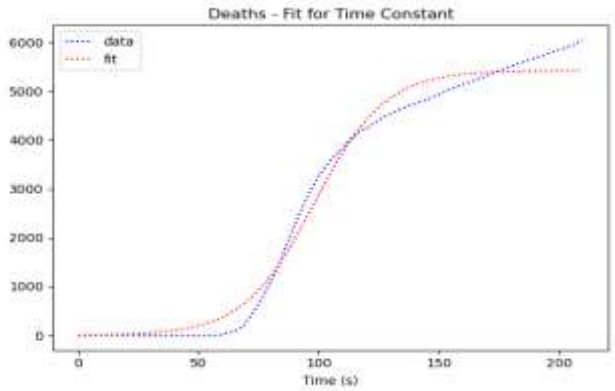
E. Algoritmaların Karşılaştırılması



Şekil 12. LSTM algoritması Hasta sayısı

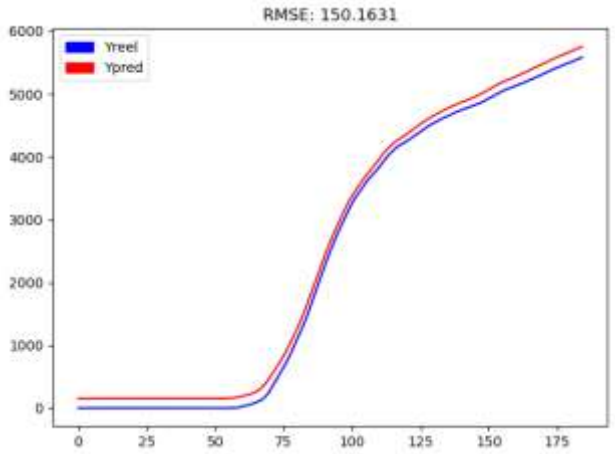


Şekil 13. Bi-LSTM algoritması Hasta Sayısı

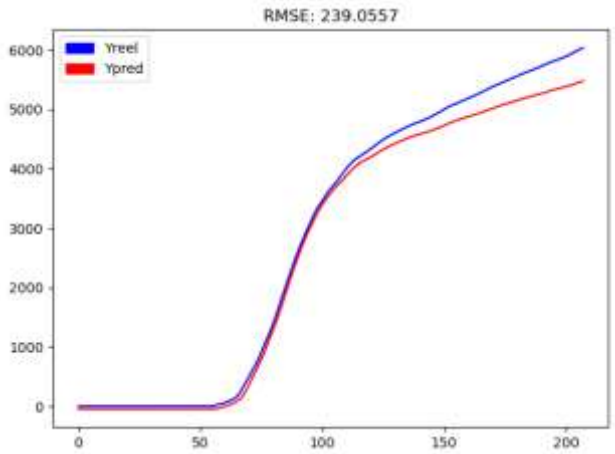


Şekil 14. Eğri Uydurma algoritması Hasta Sayısı

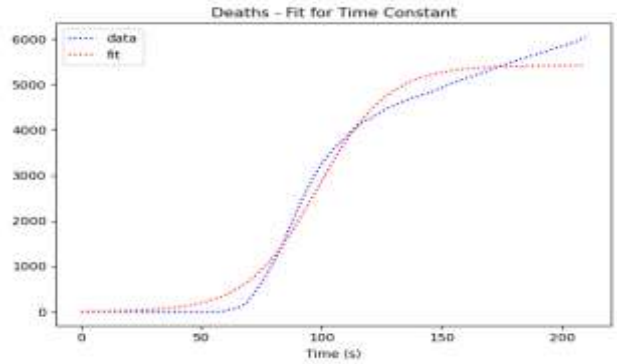
Çıkan Sonuçları karşılaştırdığımızda hasta sayısı verilerinde bize en iyi sonucu veren algoritmanın, LSTM algoritması olduğuna ulaşmış olduk.



Şekil 15. LSTM algoritması Ölüm Sayısı

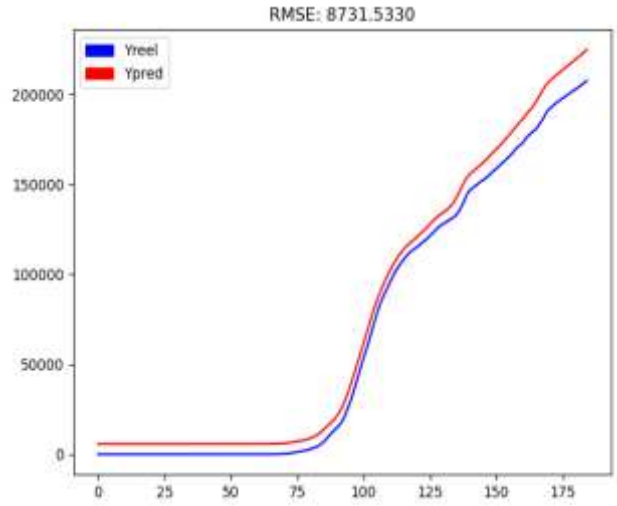


Şekil 16. Bi-LSTM algoritması Ölüm Sayısı

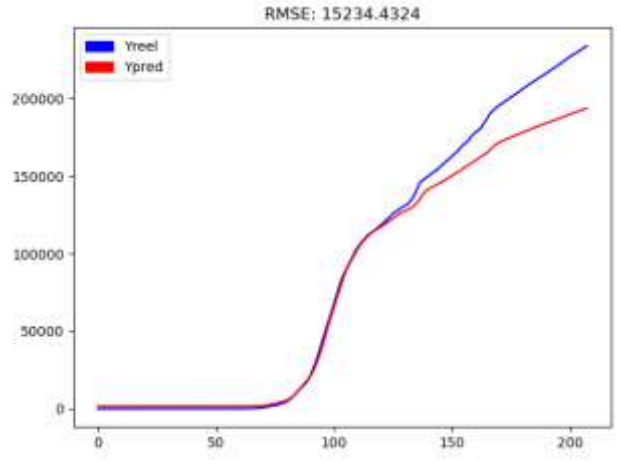


Şekil 17. Eğri Uydurma algoritması Ölüm Sayısı

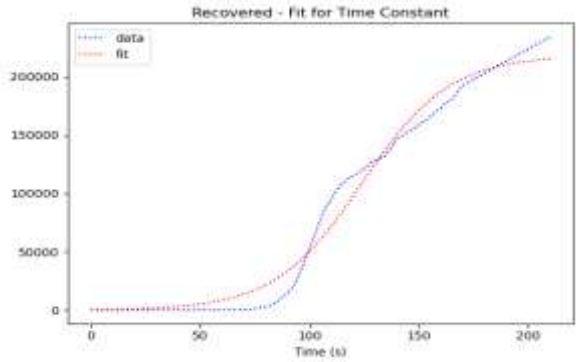
Çıkan Sonuçları karşılaştırdığımızda ölüm sayısı verilerinde bize en iyi sonucu veren algoritmanın, LSTM algoritması olduğuna ulaşmış olduk.



Şekil 18. LSTM algoritması İyileşen Sayısı



Şekil 19. Bi-LSTM algoritması İyileşen Sayısı



Şekil 20. Eğri Uydurma algoritması İyileşen Sayısı

Çıkan Sonuçları karşılaştırdığımızda iyileşen sayısı verilerinde bize en iyi sonucu veren algoritmanın, yine LSTM algoritması olduğuna ulaşmış olduk.

IV. SONUÇLAR

Uzun kısa süreli bellek (LSTM), Eğri Uydurma ve çift yönlü uzun kısa süreli bellek modelleri oluşturularak algoritmaların performansları Covid-19 Türkiye verileri üzerinde analizi yapılmıştır. Çalışmalardaki bulgular şu şekilde çıkmaktadır;

Oluşturulan tüm modellerde gerçek veriler ile eğitilen verilere en yakın çıkan algoritma LSTM algoritması olarak karşımıza çıkmıştır.

Ortalama karekök hatası (RMSE) sonuçlarına göre LSTM algoritması en düşük sonuçları vermektedir.

Süre olarak en uzun süren algoritma LSTM algoritması olarak sonuç vermiştir. En hızlısı ise eğri uydurma algoritması olarak karşımıza çıkmıştır.

KAYNAKÇA

[1] https://covid19.saglik.gov.tr

[2] "Convolutional Neural Network (ConvNet yada CNN) nedir, nasıl çalışır?", https://medium.com/@tuncerergin

[3] "A Beginner's Guide to LSTMs and Recurrent Neural Networks | Skymind", pathmind.ai, 2018. [Çevrimiçi]. Available at: https://pathmind.ai/wiki/lstm

[4] M.A KIZRAK, B BOLAT "Derin öğrenme ile kalabalık analizi üzerine detaylı bir araştırma", Bilişim Teknolojileri Dergisi, 2018

[5] https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning/

[6] T. Flow, https://www.tensorflow.org/

[7] Keras, https://keras.io/